

University of Groningen

## Discrete-Event Systems Modeling and the Model Predictive Allocation Algorithm for Integrated Berth and Quay Crane Allocation

Tri Cahyono, Rully; Flonk, Engel; Jayawardhana, Bayu

*Published in:*  
IEEE transactions on intelligent transportation systems

*DOI:*  
[10.1109/TITS.2019.2910283](https://doi.org/10.1109/TITS.2019.2910283)

**IMPORTANT NOTE:** You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

*Document Version*  
Early version, also known as pre-print

*Publication date:*  
2019

[Link to publication in University of Groningen/UMCG research database](#)

### *Citation for published version (APA):*

Tri Cahyono, R., Flonk, E., & Jayawardhana, B. (2019). Discrete-Event Systems Modeling and the Model Predictive Allocation Algorithm for Integrated Berth and Quay Crane Allocation. *IEEE transactions on intelligent transportation systems*, 21(3), 1321-1331. <https://doi.org/10.1109/TITS.2019.2910283>

### **Copyright**

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

### **Take-down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

# Discrete-event systems modeling and model predictive allocation algorithm for integrated berth and quay crane allocation\*

Rully Cahyono, Engel Flonk, and Bayu Jayawardhana, *Senior Member, IEEE*

**Abstract**—In this paper, we study the problem of integrated berth and quay crane allocation (I-BCAP) in general seaport container terminals and propose model predictive allocation (MPA) algorithm and preconditioning methods for solving I-BCAP. Firstly, we propose a dynamical modeling framework based on discrete-event systems (DES) that describes the operation of berthing process with multiple discrete berthing positions and multiple quay cranes. Secondly, based on the discrete-event model, we propose a MPA algorithm for solving I-BCAP using model predictive control (MPC) principle with a rolling event horizon. The validation and performance evaluation of the proposed modeling framework and allocation method are done using: (i). extensive Monte-Carlo simulations with realistically-generated datasets; (ii). real dataset from a container terminal in Tanjung Priuk port, located in Jakarta, Indonesia; and (iii). real life field experiment at the aforementioned container terminal. The numerical simulation results show that our proposed MPA algorithm can improve the efficiency of the process where the total handling and waiting cost is reduced by approximately 6 - 9% in comparison to the commonly adapted method of first-come first-served (FCFS) (for the berthing process) combined with the density-based quay cranes allocation (DBQA) strategy. Moreover, the proposed method outperforms the state-of-the-art HPSO-based and GA-based method proposed in recent literature. The real life field experiment shows an improvement of about 6% in comparison to the existing allocation method used in the terminal.

## I. INTRODUCTION

NOWADAYS, more than 60% of sea-cargo is transported in containers and the development of a standardized containerization system has marked the era of container terminals [25]. High cost incurred in the operation processes and stiff competition among terminals have pushed terminal operators to improve their service. Two options that are commonly looked at are 1) investment in additional equipments; and 2) improvement of the operation of the current process. In this paper, we will study the latter option. A common container terminal layout is shown in Figure 1.

An important process in the terminal operations is the seaside operations-level decision making for the berth and

quay cranes allocation. The assignment of berth positions and quay crane (QC) to incoming ships for handling their cargo plays an important role in minimizing the turnaround time. We refer interested readers to the papers [3] and [9] for an extensive review on the berth allocation problem and its current available solutions.

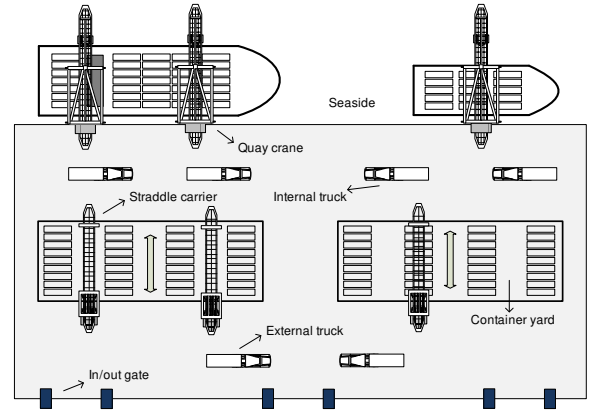


Fig. 1. An illustration of a container terminal layout with multiple berthing positions and multiple QC.

In the current literature, there are mainly two classes of berth allocation (BAP) problems which are based on the way they model the ship arrivals. These classes are the static BAP and the dynamic BAP, which is known as the DBAP as reviewed recently in [3] and [9]. In the former problem, as presented in [4], [15], [16], and [27], the entire arriving ships are assumed to have arrived at the port when the planning for the entire time interval is being made. Hence the berth allocation problem is solved based on a static set of ships' arrival time. First come first served (FCFS) rule, which is the most common method in allocating berth positions and cranes, is also based on the same assumption. It has been known that the FCFS method, which is simple and is easily adopted to the incoming ships, is not always efficient and applicable [19]. For instance, the FCFS can not be used if there is priority in seaport service where some ships can have higher priority than the others.

As opposed to the static set of ships' arrival time, the DBAP takes into account the uncertainty in the arrival time of the ships within the planning horizon [17], [19], [20], [22], and [18]. Although the ships' arrival time is allowed to vary (hence the "dynamic" term is used), the berthing allocation method in these works is based on recasting the time-varying problem to a (mixed-integer) linear programming where the uncertainties is defined as stochastic constraint; similar to the ones used in [17]-[22]. As another example, in [19], the model of the berthing process comprises of a set of linear equations (without dynamics) and inequalities that describe physical bounds as

\*This work was supported by Excellence Scholarship from Directorate General of Higher Education of Republic of Indonesia, Indonesia Port Corporation (IPC) and IPC Corporate University under the project "Analysis and design of automated distributed container terminal operations systems".

Rully Cahyono is with Department of Industrial Engineering, Faculty of Industrial Technology, Institut Teknologi Bandung, Jalan Ganesha 10, Bandung, 40132, Indonesia. e-mail: rully@ti.itb.ac.id.

Engel Flonk is a former graduate student in Industrial Engineering and Management, University of Groningen. e-mail: e.j.flonk@student.rug.nl.

Rully Cahyono and Bayu Jayawardhana are with Research Group Discrete Technology and Production Automation, Faculty of Science and Engineering, University Groningen, Nijenborgh 4 9747 AG, Groningen, the Netherlands. e-mail: r.tri.cahyono@rug.nl, b.jayawardhana@rug.nl.

well as uncertainty variables which represent the variation of the ships' arrival time. The model in [19] does not incorporate dynamical equations that describe the dynamics of the berthing operations and does not use the available real-time information of the arriving ships. Consequently the resulting allocation is conservative as it has to deal with the prescribed uncertainties on the ships' arrival and it does not feed back the real-time factual information of the arriving ships. For enabling a real-time allocation method that can handle a dynamically changing environment, a simple (yet useful) dynamical model is needed that can capture the essential elements in the terminal operations dynamics and be applicable for the development of optimal predictive allocation methods.

As our first contribution, we propose a dynamical modeling framework using a discrete-event system (DES) formulation that describes both the real-time and continuously changing set of ship arrivals at any given time, as well as, the discrete-event dynamics during the berthing and loading/unloading process. The DES formulation fits better to terminal operations than the usual periodic discrete-time systems description since there is aperiodicity in the ships' arrival time and the operations' time among different berthing positions is usually asynchronous. The book [10] provides an excellent exposition to the modeling and analysis of DES. In [13] such DES modeling framework is used to describe manufacturing and transportation systems. While in [6], the event is triggered every time uncertainty occurs in the terminal. Our proposed modeling framework fits well with the common practice in the terminal operations. Firstly, the berth planning is done based on the pro forma windows of incoming ships where the information may be incomplete and will change during the execution window. In the current state-of-the-art OR modeling framework, such uncertainty is embedded in the constraints and introduces sub-optimality in the solution. In our framework, the dynamical modeling of ships' arrival allows for a real-time planning according to real-time factual information from the arriving ships. Secondly, the state equations (which are given by difference equations) capture the sequential process in seaside operations to a large extent and is also validated later in our real life experiment. Thirdly, the simple model allows us to not only gain insight to the sequential process, but also to deploy it in our real-time integrated allocation algorithm.

A closely related problem to BAP in the terminal operations is the crane allocation problem (CAP). While BAP is related to the allocation of incoming vessels to specific berth positions, CAP deals with the allocation and scheduling of QC to the already-assigned berthing ships. Until now, there have been works that focus on the integration of the BAP and CAP, such as in [11], [14], [20], [28], and [24] which is known as berth and crane allocation problem (BCAP). But, there are two main common limitations in the current literature that leads to sub-optimal solution in practice. The first limitation is related to their inability for handling real-time factual information that differs from the apriori information used for the planning, as discussed before. The second limitation is that they solve BAP and CAP in separate (but sequential) steps, due to complexity in the problem. The first step is solving the BAP without considering the CAP. The next step is allocating QC to the

already-assigned ships.

As reviewed in [9], the BAP falls into NP-hard problems because of its complexity. The complexity itself is caused by the dimension of the problem i.e. the number of ships, the number of berth positions, and the number of quay cranes. One of the popular methods in solving the NP-hard problems, including the BAP, is genetic algorithm (GA) [9]. The GA allows flexibility for its users to solve the original problem through GA specific algorithm. The GA is employed in [11] and [20]. We will use a GA-based method as a benchmark for our optimization algorithm.

Another technique to solve the BAP is Tabu search as in [14] where the objective function is to minimize the housekeeping cost that is affected by the resulting ship schedule. While in [28], Lagrangian relaxation is used.

For the CAP, the same techniques as mentioned above are not always used. This is due to the fact that the BAP and CAP are not solved simultaneously. Therefore, the method used to solve BCAP can vary. For instance, in [20], GA is used to solve the BAP, but the same method cannot be applied to solve the CAP because of GA limitation. Therefore, in [20] the CAP is modified into a maximum flow problem-based algorithm. While in the works of [11], [14], and [28], GA, a mixed-integer linear programming (MILP) and sub-gradient method are used, respectively.

A recent work on the integrated berth and quay-crane allocation is the paper [18]. In [18], the authors propose the use of Particle Swarm Optimization, which is another nature-inspired computational tools, to solve the integrated allocation problem. The inclusion of event-based berth plan which is able to incorporate the changing in ship arrivals is one of the main contribution of [18].

The non-robustness and non-adaptiveness of the above mentioned approaches to the dynamically changing environment has led to the wide adoption in real-life condition of a very simple heuristic approach, that is a combination of the first-come first-served strategy for the berthing allocation and of the density-based strategy for the quay crane allocation.

In order to handle such dynamic environment (including the real-time information on arriving ships), we propose a novel real-time integrated berthing and crane allocation method in the present paper as our second contribution of the paper which is based on model predictive control principle and rolling horizon implementation. Finally, as our last contribution, we evaluate the performance of our model predictive allocation strategy using: (i). extensive Monte-Carlo simulations using realistic datasets; (ii). real dataset from a container terminal in Tanjung Priuk port, Jakarta, Indonesia; and (iii). real life field experiment in the aforementioned container terminal. To the best of our knowledge, the latter contribution on the real life field experiment provides an important insight to the performance of novel allocation method in reality which is typically not reported in literature.

Our simulation results show significant improvement of up to 20% (based on a single actual dataset) or up to 9% (based on realistic large-scale Monte Carlo simulations) in comparison to the traditional method of FCFS & DBQA. More importantly our method outperforms the current state-of-the-art method as

in [20] and [18]. On the other hand, the field experimental result shows an improvement of about 6% in comparison to the existing allocation method (based on the use of validated model). We have also observed from the field experiment that there is a minor difference between the validated model and data from the actual berthing process which shows that our dynamical model has not yet captured all elements in port operations. As a result, it decreases the predictive capability of the model and limits the potential of our model predictive allocation method. We will discuss possible improvement of this in future works as presented in Section VI.

## II. DYNAMICAL MODELING OF BERTHING PROCESS

In this section, we explain a generic dynamical model of berthing process in general seaport container terminals. First, in order to simplify the presentation, let us consider a simple berthing problem where there is only one berthing position and one QC. In this particular case, the decision variable is the berthing ship that is chosen from the set of ships-ready-to-be-berthed. In the generalization of this simple problem to the multiple berthing positions and multiple QC, we consider also the number of QC per berthing position as an additional decision variable. We summarize the notations in our modeling framework in Table I.

### A. The dynamic modeling of a simple berthing process

Before we start describing the berthing process dynamics, let us briefly recall the concept of DES which is a class of dynamical systems as expounded in [10]. Generically, DES are characterized by a *discrete* set of state space whose state transition is driven by (asynchronous discrete) events over time. Such class of systems encompasses systems described by automata and petri nets and it includes queueing systems, traffic systems, communication systems, etc. We refer interested readers to the book [10].

Using such DES formalism, let us introduce the event time  $k \in \mathbb{N}$  as a discrete sequence of events that corresponds to every initiation of a berthing process (at any berthing position). Thus, each event time  $k$  is related one-to-one to a unique time instance when a berthing process commences and such relation is denoted later by  $t_{bst}(k)$ . We denote  $\mathcal{S}(k)$  as the set of arriving ships to seaport at event time  $k \in \mathbb{N}$ . Here, arriving ships refer to all ships that have already reported to the port on their incoming. We denote the  $i$ -th element of  $\mathcal{S}(k)$  by  $\mathcal{S}_i(k)$ . For instance, using the actual set of ships from our experimental dataset (see also Table III in Section IV), the set  $\mathcal{S}(k)$  can be  $\mathcal{S}(k) = \{\text{"Berlian"}, \text{"Fatima"}, \text{"Meratus I"}\}$  and consequently  $\mathcal{S}_2(k)$  refers to "Fatima".

Associated to  $\mathcal{S}(k)$ , we define two different measures,  $\mu_a$  and  $\mu_o$  which correspond to the arrival time and operations time, respectively. As an illustration using the above example of  $\mathcal{S}(k)$ ,  $\mu_a(\mathcal{S}_2(k))$  refers to the arrival time of the ship "Fatima", as it is the second element in the set. Similarly,  $\mu_o(\mathcal{S}_1(k))$  and  $\mu_o(\mathcal{S}(k))$  refer to the operation time of the ship "Berlian" and to the total operation time of all ships in  $\mathcal{S}(k)$ , respectively. Here, the measure  $\mu_o(\mathcal{S}_i(k))$  refers to the  $i$ -th ship operations time for unloading and loading the entire

TABLE I  
LIST OF MATHEMATICAL NOTATIONS

Notation	Description
<b>Decision variables</b>	
$u(k)$	a control variable of the ship to be berthed chosen from the set $\mathcal{S}(k)$
$v_b(k)$	a control variable of the number of QC allocated to the $b$ -th berth position
<b>Parameters</b>	
$\mu_a$	Measure defined on elements of $\mathcal{S}(k)$ that gives the arrival time of the element(s)
$\mu_o$	Measure defined on elements of $\mathcal{S}(k)$ that gives the operations time of the element(s)
<b>State variables</b>	
$t_{bst}(k)$	the state variable of berth starting time for the simple BCAP
$t_{qcs}(k)$	the state variable of QC operations time for the simple BCAP
$t_{fin}(k)$	the state variable of berth finishing time for the simple BCAP
$z_b(k)$	the state variable of berth starting time of the $b$ -th berth position for the complex BCAP
$y_b(k)$	the state variable of remaining operations time of the $b$ -th berth position for the complex BCAP
$x_b(k)$	the state variable of finishing time of the $b$ -th berth position for the complex BCAP
<b>Sets and indices</b>	
$k$	event time
$\mathbb{N}$	the set of natural numbers
$\mathcal{S}(k)$	a dynamic set of arriving ships to seaport at event time $k$
$\mathcal{S}_i(k)$	The $i$ -th element (ship) in the set $\mathcal{S}(k)$
$\mathcal{B}$	The discrete set of berthing positions
$ \mathcal{B} $	The cardinality of the set $\mathcal{B}$
$\mathcal{Q}$	The discrete set of quay cranes
$ \mathcal{Q} $	The cardinality of the set $\mathcal{Q}$
$j$	Index of the first earliest available berth position
$b$	Index of the other berth positions, where $b \neq j$
$N$	The planning horizon
$M$	The dimension of $\mathcal{S}(k)$ where $M > N$

containers by a single QC. This choice will be useful later when we take the number of QC as another decision variable. The total operations time itself depends on the number of containers in a ship, represented by twenty-foot equivalent unit (TEU), number of QC assigned to the ship, and quay crane capacity, that is usually in TEU per hour.

As one of the decision/input variables, we denote  $u(k)$  as the ship to be berthed chosen from the set  $\mathcal{S}(k)$ . By defining  $t_{bst}(k)$  as the berth starting time,  $t_{qcs}(k)$  as the QC operation time and  $t_{fin}(k)$  as the berth finishing time, the state space equation of the berthing process at a given event time  $k$  can be given by

$$t_{bst}(k) = \max\{\mu_a(u(k)), t_{fin}(k-1)\} \quad (1)$$

$$t_{qcs}(k) = \max\{t_{bst}(k), t_{fin}(k-1)\} \quad (2)$$

$$t_{fin}(k) = t_{qcs}(k) + \mu_o(u(k)) \quad (3)$$

$$\mathcal{S}(k) = \mathcal{S}(k-1) \setminus u(k) \cup \mathcal{U}(k).$$

The state space for  $t_{bst}$ ,  $t_{qcs}$  and  $t_{fin}$  is  $\mathbb{N}$  and for  $\mathcal{S}$  is the discrete set of all (admissible) arriving ships to the port. One

can further simplify (1)-(3) into only two state equations as follows

$$x(k) = \max\{\mu_a(u(k)), x(k-1)\} + \mu_o(u(k)) \quad (4)$$

$$\mathcal{S}(k) = \mathcal{S}(k-1) \setminus u(k) \cup \mathcal{U}(k) \quad (5)$$

where the state  $x(k)$  denotes the finishing time  $t_{\text{fin}}(k)$  and  $\mathcal{U}(k)$  is the set of new arriving ships that comes at the event time  $k$ . In (5), we have that the set of incoming ships at each time  $k$  is the same set from the previous time step modulo (c.f. the symbol  $\setminus$ ) the ship that has been taken out from the set for berthing (e.g., the ship  $u(k)$ ) and is added by (c.f. the symbol  $\cup$ ) a (possible) set of incoming ship(s)  $\mathcal{U}(k)$  arriving at time  $k$ .

*Remark 1:* Compared to the existing literature ([1], [2], [5]), our dynamical modeling framework for the berthing problem has resulted into state equations involving set dynamics (c.f. (5)). The analysis of such dynamics in the context of seaports interconnection is not trivial and we will not treat this issue in this paper. However, we still take into account the set dynamics in our optimization problem later.

### B. Generalization to the multiple berthing positions and multiple QC

In this subsection, we will extend the dynamical modeling of a simple berthing process in (4)-(5) into multiple berth positions and multiple QC. For defining the domain of our decision variables, we denote the set of discrete berthing positions by  $\mathcal{B}$  where  $|\mathcal{B}|$  is the total number of positions and we denote  $\mathcal{Q}$  as the set of QC where  $|\mathcal{Q}|$  defines the total number of available QC. Note that we consider only discrete berthing positions in this paper. We denote  $x_b(k)$  as the finishing time of the  $b$ -th berth position at the event time  $k$ , for all  $b = 1, \dots, |\mathcal{B}|$ .

In this setting, every time an assigned QC has finished an operation at a particular berth position, a new berthing process will commence where a new ship needs to be allocated and berthed. This means that this is an event-based dynamical model, since  $k$  is triggered from a completed event from previous  $k-1$ . As before, the finishing time for the  $b$ -th position will be denoted by  $x_b(k)$ .

To capture the complexity, in contrast to the dynamical modeling for the simple berthing process, we need at least three state variables for every berthing position that record the starting time  $t_{\text{bst}}$ , the estimated finishing time  $t_{\text{fin}}$  and the remaining operations time. For every  $b$ -th berth position, we denote new state variables  $z_b(k)$  as the berth starting time and  $y_b(k)$  as the remaining operations time. We define an additional control variable  $v_b(k) \in \mathbb{N}$  which is the number of QC allocated to the  $b$ -th berth position at step  $k$  and we assume that the total number of QC is constant during the entire operations. For every event time  $k$ , the dynamics is given as follows. By letting

$$j = \arg \min_b [x_b(k-1)] \quad (6)$$

the dynamics of the  $j$ -th berth position is given by

$$z_j(k) = \max\{x_j(k-1), \mu_a(u(k))\} \quad (7)$$

$$y_j(k) = \mu_o(u(k)) \quad (8)$$

$$x_j(k) = z_j(k) + \frac{y_j(k)}{v_j(k)} \quad (9)$$

and the dynamics of the other berth positions  $b \neq j$  is given by

$$z_b(k) = \begin{cases} x_j(k-1) & \text{if } x_j(k-1) > z_b(k-1) \\ z_b(k-1) & \text{otherwise} \end{cases} \quad (10)$$

$$y_b(k) = y_b(k-1) \quad (11)$$

$$- [z_b(k) - z_b(k-1)]v_b(k-1)$$

$$x_b(k) = z_b(k) + \frac{y_b(k)}{v_b(k)} \quad (12)$$

$$\sum_{b=1}^{|\mathcal{B}|} v_b(k) = |\mathcal{Q}| \quad (13)$$

$$\mathcal{S}(k) = \mathcal{S}(k-1) \setminus u(k) \cup \mathcal{U}(k). \quad (14)$$

Equation (6) refers to the earliest available berth position (denoted by  $j$ ) based on the finishing time of each berth position at the previous event time  $k-1$ . The state variable  $z$  in (7) and (10) defines the berthing time for every berth position at every event time  $k$ . The state variable  $y$  in (8) and (11) is the remaining operations time at every berth positions. Finally, as in the simple berthing process, the state variable  $x$  describes the estimated finishing time for every berth position based on the allocated QC given by the input variable  $v$ . The equation (13) ensures that the total number of QC assigned to all berth positions is the same as the total number of available QC.

An illustration of the complex berthing process is provided in Figure 2. In this figure, two berthing positions are considered namely, the  $j$ -th and  $b$ -th berth positions. As discussed above, at the event time  $k$ , the  $j$ -th and the previously allocated QC become available since the berthing operations has been completed. Hence, a new allocation process is started where a new ship  $u(k)$  is allocated to the vacant berth position. Simultaneously, the QC can be redistributed during the event time  $k$  which can result in the changes to the remaining operation time for the other berthing positions (e.g. the  $b$ -th berth position in this figure). We can notice from this figure that the berthing process is an asynchronous process where we cannot define a periodic time sampling as commonly used for modeling dynamical systems. Instead we use the event time  $k$ .

From the state equation (9) and (12), one can deduce that the domain of the state space is

$$\{(x, y, z) \in \mathbb{R}_+^{|\mathcal{B}|} \times \mathbb{R}_+^{|\mathcal{B}|} \times \mathbb{R}_+^{|\mathcal{B}|} \mid x_i > z_i, i = 1, \dots, |\mathcal{B}|\}.$$

It can also be seen from (7) and (10) that the state trajectories of the berthing time  $z$  is monotone non-decreasing.

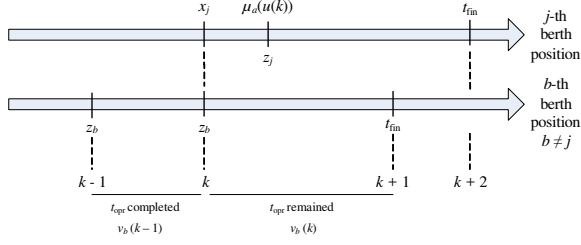


Fig. 2. Illustration of the discrete-event systems in the berthing process with multiple berthing positions and multiple QC. It shows the dynamical relationship between two berthing positions at each event time  $k$ . The variables  $x_j$ ,  $z_j$ ,  $x_b$  and  $z_b$  are as defined in (7)–(12).

### III. MODEL PREDICTIVE ALLOCATION STRATEGY

In this section, we propose a model predictive allocation strategy for solving the I-BCAP where we modify the standard model predictive control approach to our discrete-event systems formulation presented in the previous section. In our proposed approach, the DES model of berthing process as presented in Section II, is used to optimize the berthing control input  $u$  and quay cranes control input  $v$  for a finite events horizon in the future and subsequently, the solution for the current event is implemented and the horizon is rolled by one event time further.

In the following, we discuss first the cost functions that will be optimized by our proposed model predictive allocation algorithm, as well as, be used for comparing our methods with existing approaches. Subsequently, we present our proposed algorithm and discuss two preconditioning steps for tractably solving the optimization problem.

#### A. Objective functions

The cost function of the berthing process, whether for the simplest one or for the multiple berth positions and QC, is based on both the operations cost and waiting cost. These two cost components are closely related to the time that the ships spent at the assigned berth positions for completing their berthing process.

The operations cost is the cost of operating the QC that are allocated to a particular ship. Let us denote the operating cost of a QC unit (€/hour) by  $C_o$ . The operational cost between the step  $k-1$  to step  $k$  is then defined by  $C_o$  multiplied by the time needed for unloading/loading containers from/to a ship. In other words, it is given by  $C_o \mu_o(u(k))$  and

$$C_o \left( [x_j(k-1) - z_j(k-1)]v_j(k-1) + \sum_{b \neq j} [z_b(k) - z_b(k-1)]v_b(k-1) \right),$$

for a single QC and multiple QC case, respectively, where  $j$  is as in (6).

On the other hand, the waiting cost is associated to the total time that a particular ship spends at seaport, i.e. from the time it arrives until it leaves after the assigned QC have completed the operations. It may happen that the particular ship has to wait after its arrival, since all berth positions are occupied. We denote  $C_w$  the waiting cost of a ship unit (€/hour).

For simplicity of notation, for the multiple berth positions and multiple QC, we denote the earliest available berth position at event time  $k$  by  $w(k)$  which is defined by

$$w(k) = \arg \min_b [x_b(k-1)]. \quad (15)$$

Based on this description, the cost functions (defined from the step  $k$  with horizon  $N$ ) for the simple berthing process and for that with multiple berth positions and multiple QC are given by

$$J(x, u) = \sum_{n=k}^{k+N} \mu_o(u(n))C_o + [x(n) - \mu_a(u(n))]C_w \quad (16)$$

and

$$\begin{aligned} J(x, y, z, u, v) &= \sum_{n=k}^{k+N} \left( [x_{w(n)}(n-1) - z_{w(n)}(n-1)]v_{w(n)}(n-1)C_o \right. \\ &\quad \left. + \max\{x_{w(n)}(n-1) - \mu_a(u(n)), 0\}C_w \right. \\ &\quad \left. + \sum_{b \neq w(n)} [z_b(n) - z_b(n-1)](v_b(n-1)C_o + C_w) \right) \end{aligned} \quad (17)$$

respectively.

In this formulation, we have explicitly defined the cost function as a function of state variables  $x, y$  and  $z$  and of input variables  $u$  and  $v$  that satisfy (6)–(14). Note that since  $x > z$  (as remarked after (14)), the cost function  $J$  in (17) is positive definite.

#### B. Model predictive allocation algorithm

Let us now describe our model predictive allocation (MPA) algorithm. For every event time  $k$ , we denote  $\hat{z}(l)$ ,  $\hat{y}(l)$ , and  $\hat{x}(l)$ , with the integer  $l \geq 0$ , as the predicted state variables at event time  $k+l$  based on known/measured state variables at current event time  $k$ . Using this notation,  $\hat{z}(0) = z(k)$ ,  $\hat{y}(0) = y(k)$  and  $\hat{x}(0) = x(k)$ . Also,  $\hat{z}(-1) = z(k-1)$ ,  $\hat{y}(-1) = y(k-1)$  and  $\hat{x}(-1) = x(k-1)$ . Similar to (15), we define  $\hat{w}(l) = \arg \min_b \hat{x}_b(l-1)$ . Using these notations, the MPA algorithm for the berth and quay cranes is given as follows where we use the event horizon  $\{0, 1, 2, \dots, N\}$  with  $N > 0$  for the predictive state variables  $(\hat{z}, \hat{y}, \hat{x})$ , which is equivalent to the rolling event horizon  $\{k, k+1, k+2, \dots, k+N\}$  for the state variables  $(z, y, x)$ . For generality, we will describe the algorithm for the multiple berthing positions and multiple QC case. It is straightforward to adapt the algorithm for the simple berthing process.

#### MPA Algorithm

- 1) For a new event time  $k$ , we update the current state variables as in (6)–(14).
- 2) Solve the following optimization problem

$$\min_{\hat{u}, \hat{v}} J(\hat{x}, \hat{y}, \hat{z}, \hat{u}, \hat{v})$$

subject to

$$\hat{z}_{\hat{w}(l)}(l) = \max\{\hat{x}_{\hat{w}(l)}(l-1), \mu_a(\hat{u}(l))\} \quad (18)$$

$$\hat{y}_{\hat{w}(l)}(l) = \mu_o(\hat{u}(l)) \quad (19)$$

$$\hat{x}_{\hat{w}(l)}(l) = \hat{z}_{\hat{w}(l)}(l) + \frac{\hat{y}_{\hat{w}(l)}(l)}{v_{\hat{w}(l)}(l)} \quad (20)$$

and for every  $b \neq \hat{w}(l)$

$$\hat{z}_b(l) = \begin{cases} \hat{x}_{\hat{w}(l)}(l-1) & \text{if } x_{\hat{w}(l)}(l-1) > z_b(l-1) \\ z_b(l-1) & \text{otherwise} \end{cases} \quad (21)$$

$$\hat{y}_b(l) = \hat{y}_b(l-1) - [\hat{z}_b(l) - \hat{z}_b(l-1)]\hat{v}_b(l-1) \quad (22)$$

$$\hat{x}_b(l) = \hat{z}_b(l) + \frac{\hat{y}_b(l)}{\hat{v}_b(l)} \quad (23)$$

$$\sum_{b=1}^{|\mathcal{B}|} \hat{v}_b(l) = |\mathcal{Q}| \quad (24)$$

$$\hat{\mathcal{S}}(l) = \hat{\mathcal{S}}(l-1) \setminus \hat{u}(l), \quad (25)$$

where  $l = 0, 1, N$ ,  $\hat{u}$  and  $\hat{v}$  are the predicted control input within the horizon.

- 3) Implement the berthing control input  $u(k) = \hat{u}(0)$  and the quay crane control input  $v(k) = \hat{v}(0)$ .
- 4) Increment the event time  $k$  by one and return to 1).

### C. Preconditioning steps

For solving the optimization problem in Step 2 of the MPA algorithm above in the event horizon  $\{0, 1, \dots, N\}$ , we need to find an optimal berthing control input  $\hat{u}(0), \dots, \hat{u}(N)$  from the space of  $\mathcal{S}(k)$  and an optimal quay cranes control input  $\hat{v}(0), \dots, \hat{v}(N)$  from the available number of quay cranes  $|\mathcal{Q}|$ . If  $|\mathcal{S}(k)|$  and  $|\mathcal{Q}|$  are small, one can easily solve the optimization problem by using an exhaustive search. When they are very large (with, for instance,  $M := \dim \mathcal{S}(k) \gg N$ ), solving such combinatoric problem is NP-hard and one can use a heuristic method, such as, the Genetic Algorithm (a popular nature-inspired computational method and is used for solving I-BCAP in [20]), for finding a (sub)-optimal sequence of  $N$  ships that minimizes the cost function.

As an alternative to GA and HPSO for solving the aforementioned optimization problem, we propose a preconditioning step, called *N-level FCFS*, which is a quasi-exhaustive search that combines FCFS and exhaustive search approaches. The approach is detailed as follows.

### N-level First-Come First-Served (N-level FCFS)

- 1) Order the set of ships-to-be-berthed at step  $k$ ,  $\mathcal{S}(k)$  based on the measure of the arrival time  $\mu_a$  such that

$$\mu_a(\mathcal{S}_1(k)) \leq \mu_a(\mathcal{S}_2(k)) \leq \dots \leq \mu_a(\mathcal{S}_M(k))$$

holds where  $M$  is the dimension of  $\mathcal{S}(k)$  and is assumed to be larger than the horizon  $N$ .

- 2) Pick the first  $N$  ships from the ordered set and denote such subset of ships as  $\mathcal{D}(k)$ .

- 3) Perform exhaustive search of optimal berthing control input  $\hat{u}$  from  $\mathcal{D}(k)$  and  $\hat{v}$  from the available number of quay cranes that solves the optimization problem in the given event horizon as above.

The above *N-level FCFS* replaces the optimization step 2 in the MPA algorithm as given before. This combination will be referred to as *MPA-FCFS* method throughout this paper. One can also propose another pre-conditioning step based on the measure of operations time  $\mu_o$  that is closely related to the size or container density of the incoming ships. This method is called *N-level Heavy-First Light-Last* *N-level HFLL*. It has the same procedure with the (*N-level FCFS*), but, the ship ordering in the step 1 is now based on the measure of the operational time  $\mu_o$  such that  $\mu_o(\mathcal{S}_1(k)) \geq \mu_o(\mathcal{S}_2(k)) \geq \dots \geq \mu_o(\mathcal{S}_M(k))$ . The ordering ensures that the first  $N$  ships that is defined as  $\mathcal{D}(k)$  in Step 2 will consist of the  $N$  heaviest ships from the current set of ships at the port  $\mathcal{S}(k)$ . The combination of *N-level HFLL* and the MPA algorithm will be referred to as *MPA-HFLL* method.

## IV. SIMULATION

To evaluate the efficacy of our proposed method, as well as, for comparing it with standard and state-of-the-art approaches, we firstly conduct numerical simulations as presented in this section. Subsequently, we present experimental validation in the next section, i.e., Section V. The simulations comprise of two different setups. In the first simulation, we use an actual dataset obtained from a real container terminal. In the second one, we use large-scale realistically generated datasets to test the robustness of our proposed method through Monte-Carlo simulations.

### A. Simulation setup

Let us describe the simulation setup for both types of datasets where we define how the data are gathered (or generated) and discuss the underlying assumptions. For the computer simulations, we use a standard personal computer with a 1.60 GHz Intel Core i7-720QM processor. The experimental data are stored in microsoft Excel and processed using VBA 2016 and Matlab version 2016a. All other data processing is done using Matlab version 2016a.

In both settings, we will compare our proposed approach with two different benchmark methods which are commonly used in practice. The first benchmark method is the combination of FCFS and density-based QC allocation (DBQA) which are also the current policies in the terminal considered in the field experiment. The DBQA allocates the QCs according to the container density of the entire ships that are currently berthed in the seaport. The second benchmark method is the GA-based solution as recently proposed in [20]. We remark that Hybrid Particle Swarm Optimization (HPSO) algorithm, which is another nature-inspired computational algorithm, has been used to solve the I-BCAP in [18]. In [18], it is reported that the performance of HPSO-based solution is comparable to the GA-based one. Therefore we compare our results with



the original GA-based solution as in [20] and the HPSO-based method as in [18].

For the first method, FCFS allocates ships based on their arrival times. The first earliest arriving ship will be allocated before the second earliest and this process is recursed until the entire ships are allocated. While for the DBQA, it allocates the number of QC based on the density of the already-berthed ships. The density of a ship is defined as the proportion of the particular ship's load to the total load of the entire ships which berth during the same period of a event time  $k$ . The number of QC allocated to each ship is proportional to its density.

For the second method, the proposed method in [20] consists of two separate procedures for solving I-BCAP. The berth allocation part is solved using genetic algorithm (GA) following the same procedure as outlined in [23]. For details on both the GA-based and the HPSO-based I-BCAP methods, we refer interested readers to the Supplemental Material in [8] and the paper [18].

We use the terminal standard for the  $C_o$ , i.e. the cost spent by the terminal operator to operate QC for loading and unloading containers and it is approximately given by €1,250 per hour. For the waiting cost of  $C_w$ , since there is no data available, we use information from the terminal, i.e. the estimated hourly cost spent by every ship to wait for the completed operation in the seaport, which is in our case is €7,500 per hour [21]. As a benchmark, we refer into [14] which states that the delay cost for a 15,000 TEU ship is about a million Euro per day. Since the ships considered in our simulations are at most below 7,000 TEU, our estimation of  $C_w$  is appropriate.

Let us now discuss the two datasets that will be used in our simulation below.

1) *First simulation setup*: For the first simulation setup, we used an actual dataset from a seaport in Tanjung Priuk, Jakarta. The data is obtained from the smallest terminal in the seaport which consists of 2 berth positions and 7 QC with the same technical specification. The data is collected from the primary source of Pelindo II from which we get the permission to use the data for an academic purpose. The time period of the dataset is from 20 January 2014 to 31 January 2014. There are 28 incoming ships to the terminal whose loads range from 327 to 2,156 TEU.

TABLE II  
A SUBSET OF AN ACTUAL DATASET FROM AN INDONESIAN SEAPORT IN  
JAKARTA USED IN THE FIRST SIMULATION SETUP.

No	Ship's name	TEU	Arrival time	Operations time (QC min)
1	"Berlian"	665	20-01-14 23:59	1,995
2	"Fatima"	713	20-01-14 23:59	2,139
3	"Meratus I"	750	21-01-14 2:20	2,250
4	"Sejahtera"	463	21-01-14 3:00	1,389
5	"Vertikal"	894	22-01-14 8:00	2,682
6	"T. Rejeki"	429	22-01-14 8:15	1,287
7	"Meratus II"	318	23-01-14 8:30	954
8	"Meratus III"	392	23-01-14 16:00	1,176
9	"Perintis"	368	23-01-14 23:00	1,104
10	"Meratus IV"	807	24-01-14 23:59	2,421

A subset of the data for the first 10 ships is shown in Table II. The measure  $\mu_a$  and  $\mu_o$  that we define in Section II-A, can be obtained from the arrival time and from the operations time (in QC min.), respectively.

2) *Second simulation setup*: For the second simulation setup, we generate large-scale realistic datasets for evaluating the efficacy of our method via Monte-Carlo simulations. In total, there are 300 different datasets, representing a combination of various different terminal settings, as well as, ship loads which are generated as follows.

For each dataset, we generate a set of 50 arriving ships according to one of the following scenarios: light load, normal load, and heavy load. The first scenario is the case where incoming ships arrive to the seaport in sparse inter-arrival times, and the loads are not high. On the other hand, the heavy load scenario is the opposite situation of the light load scenario.

The ship arrivals data is generated based on log-normal distribution to avoid negative value of inter-arrival times. We obtain the parameters from real data as described in Section IV-B, where from the 29 ship arrivals, the log-normal mean and standard deviation are 5.96 and 0.63 hours, respectively. Based on interviews with the operators who work at the seaport of Tanjung Priuk from which the data were taken, the arrivals can be categorized as a light one [21]. We also generate the ships loads (TEUs) based on the uniform distribution. The basis to categorize ships loads is derived from common container vessels classifications. The feeder ships' capacity are usually up to 3,000 TEUs. The Panamax ships are up to 5,000 TEUs. While the Post-Panamax is a generation of ships that are able to carry 10,000 TEUs. The important assumption is each of berth position can handle all kind of ships regardless of their sizes. Hence, we generate our parameters as follows in Table III.

TABLE III  
THE SETTING OF SIMULATION SCENARIO.

Scenario	Mean	Std. dev.	Lower bound load (TEU)	Upper bound load (TEU)
Light load	6.0	0.6	1,000	3,000
Normal load	5.5	0.5	3,000	5,000
Heavy load	5.0	0.4	5,000	10,000

For each different ship load setting, we test our method with four different terminal settings. The number of berth positions in each setting are 2, 3, 4, 5, respectively, while the QC numbers are 5, 7, 9, and 11, respectively. In all of these types of terminals, the technical specifications of the berth and QC are all the same.

## B. Simulation results

In this section, we provide simulation results for the multiple berthing positions and multiple QC case.

1) *First simulation results*: Using the first simulation setup as presented before, we apply  $N$ -level FCFS and  $N$ -level HFL pre-conditioning step to our proposed MPA method and



the simulation results are shown in Table IV where the event horizon length  $N$  is taken between 1 and 8. In these tables we provide also results using the two benchmark methods as described in Section IV-A.

TABLE IV  
SIMULATION RESULT OF THE BERTH AND QUAY CRANE ALLOCATION FOR MULTIPLE BERTH POSITIONS AND MULTIPLE QC USING OUR PROPOSED MPA WITH  $N$ -LEVEL FCFS PRE-CONDITIONING STEP.

$N$	Allocation Strategy	Total cost MPA-FCFS	Total cost MPA-HFLL
1	FCFS & DBQA	3,921,923	3,921,923
2	MPA-FCFS	3,828,642	3,883,197
3	MPA-FCFS	3,642,352	3,775,005
4	MPA-FCFS	3,547,157	3,619,357
5	MPA-FCFS	3,364,535	3,586,977
6	MPA-FCFS	3,230,941	3,434,669
7	MPA-FCFS	3,162,912	3,370,241
8	MPA-FCFS	3,115,403	3,308,427
	GA-based method	3,445,444	3,445,444
	HPSO-based method	3,246,945	3,246,945

As shown in Table IV, the total cost using our proposed MPA method monotonically decreases as the event horizon length  $N$  increases. The MPA with  $N$ -level FCFS and  $N$ -level HFLL with a horizon of 8 can already decrease the total cost of 20.56% and 15.64%, respectively, compared with the traditional FCFS and DBQA methods. In comparison to the benchmark from [20], the MPA method with 8-level FCFS and 8-level HFLL pre-conditioning result in cost reduction of 9.58% and 3.98%, respectively.

To show the effect of these various allocation methods, we present in Figure 3(a)-(d) the allocation results using the FCFS & DBQA, our proposed MPA-FCFS method with  $N = 8$ , the HPSO-based I-BCAP method as in [18], and the GA-based I-BCAP method as proposed in [20], respectively.

The small box in the right-upper part of every ship's schedule box represents number of QC assigned to the particular schedule. We can observe from these figures that the different methods can produce different berthing and QC allocation. For instance, it can be seen from Figure 3(a) which shows the allocation result using FCFS & DBQA, that "Meratus I" is berthed prior to "Sejahtera" which conforms to the arrival data as given in Table II. On the other hand, our proposed method, which gives up to 20% cost reduction as shown in Table IV, gives priority to "Sejahtera" over "Meratus I". Surprisingly, the GA-based method yields a similar allocation strategy as the FCFS & DBQA, as shown in Figure 3(d), although there are differences for the higher event time  $k$  not shown in the figures.

2) *Second simulation results:* We simulate each of 300 datasets according to discrete-event system as in (6) – (14). For each dataset, we evaluate the performance of the FCFS & DBQA, MPA with  $N$ -level FCFS, as well as, with  $N$ -level HFLL pre-conditioning steps (with  $N = 8$ ), the GA-based and HPSO-based I-BCAP methods.

The results for the light, normal, and heavy load scenario are shown in Figure 4, where we present a summary of cost reduction for each berth and quay crane configuration from 100 simulations each with respect to the *de-facto* method of

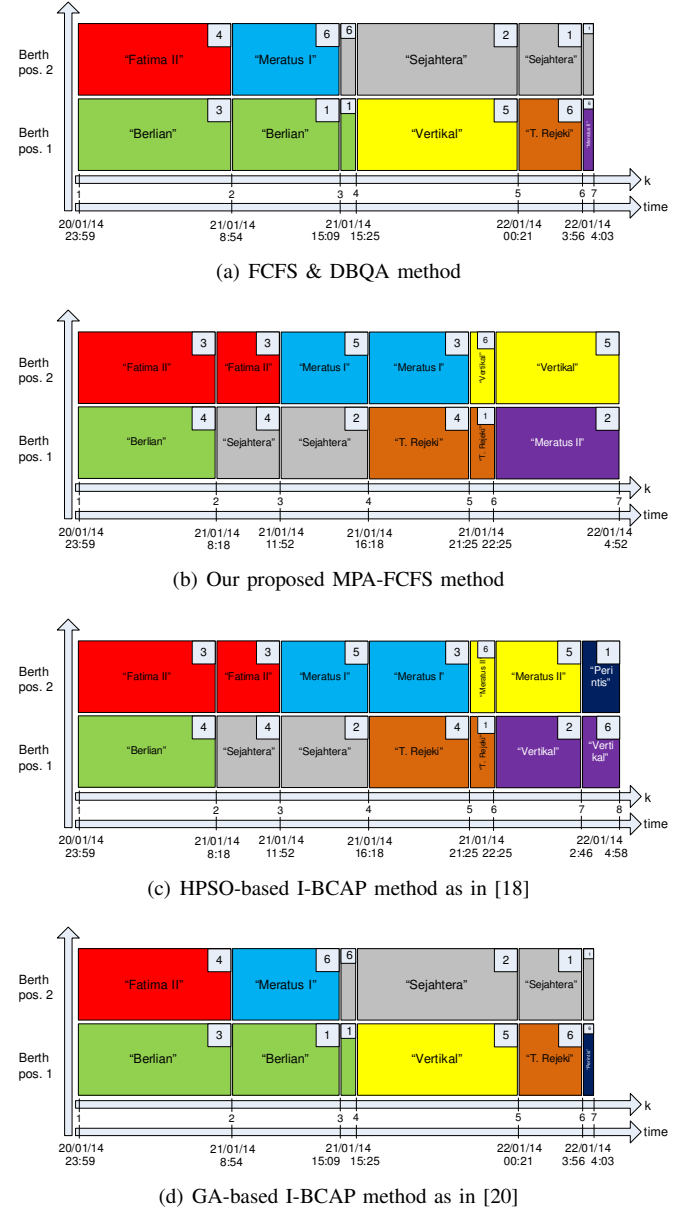
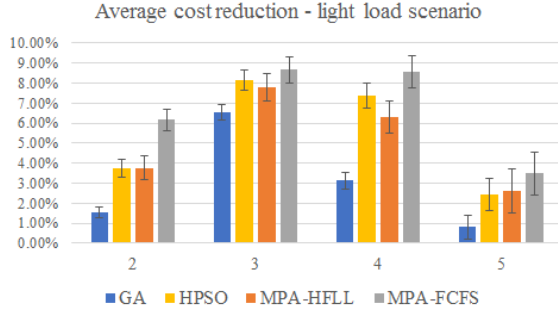


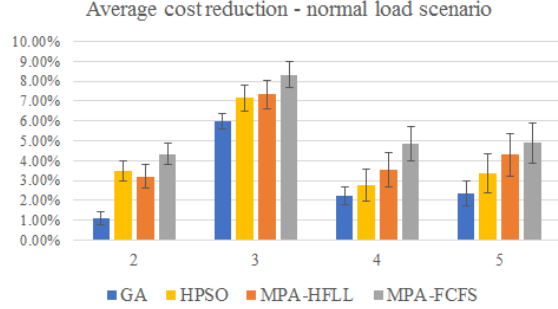
Fig. 3. The resulting berth and quay crane allocation for the first 7 event time using: (a). the de-facto FCFS and DBQA method; (b). our proposed MPA-FCFS method; (c). HPSO-based I-BCAP method as in [18]; and (d). GA-based I-BCAP method as proposed in [20]. The berth positions are shown in the vertical axis and the actual real time (and the event time  $k$ ) is shown side-by-side in the horizontal axis.

FCFS & DBQA (in percentage). The standard deviation is also given for each method and each scenario in these figures.

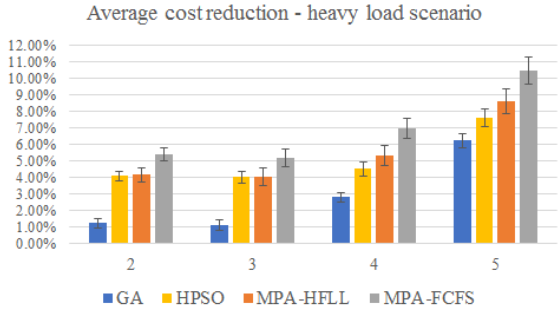
From these results we can see that for the entire scenarios, our proposed MPA with both pre-conditioning steps outperform the traditional method, as well as, when compared to the benchmark method as in [20]. Note that in all of these simulations, the ships in each scenario are not necessarily the same, because the datasets for each simulation run is different. We also present the average calculation time for each method and for each scenario in Figure 5.



(a) Light load scenario



(b) Normal load scenario



(c) Heavy load scenario

Fig. 4. Average BCAP cost reduction from 100 datasets for each berth and QC configuration with (a). light load scenario; (b). normal load scenario; and (c). heavy load scenario. The horizontal and the vertical axis are the seaport configuration and the average reduction cost, respectively. The cost reduction is calculated as a percentage from the traditional FCFS&DBQA method. Vertical line at each average cost point gives the standard deviation.

## V. FIELD EXPERIMENT

In this section, we present field experimental results using our proposed method to a real container terminal in Tanjung Priuk, Jakarta. The main purpose of the experiment is to validate our approach in a real-life field experiment. We first discuss the experimental setup in Section V-A and then present the experimental result in Section V-B.

### A. Experimental setup

The experiment was conducted in Port of Tanjung Priuk, Jakarta for one week in 2016. Indonesia Port Corporation (IPC) as the port owner as well as the operator in the terminal that we study, has given us permission to do the real life field experiment for an academic purpose. In this paper, to

Average calculation time - normal load scenario

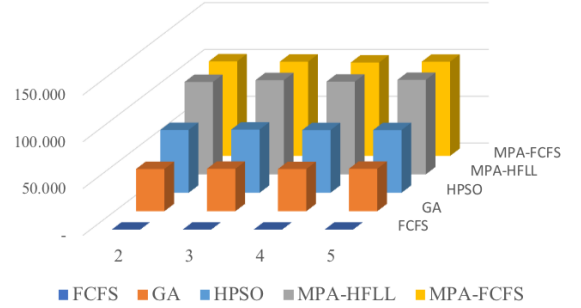


Fig. 5. The average calculation per time-step of each method for normal load scenario. The horizontal and vertical axis are seaport configuration and calculation time in seconds, respectively.

protect the identity of shipping liners, we anonymize the ship names, as well as, the time information where we initialize the time according to the first arriving ship and maintain the information on the inter-arrival time and on the load information. The original data are available upon request.

We study the arrivals of ships to a terminal which has two berth positions and seven QC, the same terminal from which the data in Section IV-A is obtained. There were, in total, eleven ships which came to the terminal as anonymously shown in Table V.

TABLE V  
ANONYMIZED DATASET OF ARRIVING SHIPS IN THE FIELD EXPERIMENT.

Ship index	Arrival date	Planned arrival time	Actual arrival time	Load (TEU)
1	Day 1	00:00	06:43	7,379
2	Day 1	17:00	17:12	5,428
3	Day 2	01:00	01:00	5,639
4	Day 3	12:00	12:00	6,988
5	Day 4	08:00	13:00	6,523
6	Day 4	11:35	11:20	4,625
7	Day 5	09:40	09:40	4,028
8	Day 5	13:00	13:50	6,853
9	Day 6	07:10	08:00	7,219
10	Day 6	17:00	17:00	7,629
11	Day 7	10:40	11:00	8,725

We can see in Table V that the arrival time is separated into two different data. The planned arrival time, known as the expected arrival time (ETA) reflects the schedule sent by each shipping liner to the terminal operators as an integrated part of container manifests and is one of the main information sources used by the terminal planner. Considering the very dynamics situation of maritime transportation, the planned schedule may alter due to many factors. The two most known factors causing delay are weather and delay from previous seaports. As a result, the actual arrival time may differ from the planned one, as given in the fourth column of the table. Ship index 5, for instance, arrived five hours late. Due to its real-time optimization capability, our proposed method can use the actual arrival time information. We note that there

is other source of uncertainties that is not taken into account in our modeling framework explicitly which has influenced the performance of our algorithm. It is the delay caused by the movement of quay cranes from one berthing position to another. Despite this, as shown in the next subsection, our algorithm can still perform very well.

### B. Model validation and experimental results

As we have a limited amount of time (one week) and as the incoming ships always vary (in terms of loads and arrival time), it is not possible to conduct field experiments using different algorithms under exact condition. Therefore, we used the whole week operations for evaluating our proposed MPA-FCFS method where the parameters in our predictive model as in (6) – (14) are based on apriori information from the bill of lading and from the real-time information. Note that the implementation of MPA-FCFS policy to the incoming ships is based on the simulation results in Section IV-B where it is shown that MPA-FCFS method performs consistently better than MPA-HFLL policy. Based on the resulting schedule using our MPA-FCFS, we then validate our predictive model (6) – (14) against the information from the actual berthing process. Figure 6 shows the evolution of one of the state variables (namely, the state of finishing time for the first berth position, denoted by  $x_1(k)$ ) based on our predictive model using the optimized ships schedule in comparison to that obtained from the actual one. In this figure, we can see clearly that our modeling framework in (6) – (14) captures well the dynamic behaviour of berthing process. The small discrepancy that is observed in this figure is mainly due to the extra dynamics introduced by the QC's movement.

Using the validated predictive model (without incorporating the delay caused by the movement of QC), we compare the performance of our MPA-FCFS with the *de-facto* FCFS & DBQA combined method, with the GA-based I-BCAP method and with the HPSO-based I-BCAP method. The results are summarized in Table VI. We calculate the cost incurred according to formula as described in Section II-A, where we used the same unit costs as provided in Section IV-A. From this table, we see that our proposed MPA-FCFS method has a lower cost of 6%, 3% and 1.8% than the standard method, the GA-based method [20] and the HPSO-based method [18], respectively. This agrees qualitatively with our simulation results discussed in the previous section. When we take into account the extra cost incurred due to the delay in moving the assets (quay cranes), it provides an explanation to the lower performance than that expected from the simulations where we can potentially gain up to 8%-9% of cost reduction (c.f. Figure 4). It is therefore foreseen that the inclusion of additional DES models that describe the dynamics of various assets in the berthing process can increase the predictive capability of such integrated models and subsequently improve the real-time optimization algorithm in our MPA method.

## VI. CONCLUSION & FUTURE WORKS

We have formulated a dynamical modeling framework of berthing process for general seaport container terminals. The

TABLE VI  
COMPARISON OF BERTH AND QUAY CRANE ALLOCATION PERFORMANCE USING THE PROPOSED MPA-FCFS METHOD, USING THE STANDARD FCFS & DBQA COMBINED METHOD AND USING THE GA-BASED I-BCAP METHOD AS IN [20]. THE MPA-FCFS METHOD IS BASED ON A FIELD EXPERIMENT IN TANJUNG PRIOK, JAKARTA, WHILE THE OTHERS ARE BASED ON A VALIDATED PREDICTIVE MODEL THAT IS USED IN THE FIELD EXPERIMENT.

Allocation Strategy	Average Total cost
FCFS&DBQA	3,921,923
MPA-FCFS	3,687,532
HPSO-based approach as in [18]	3,754,765
GA-based approach as in [20]	3,812,345

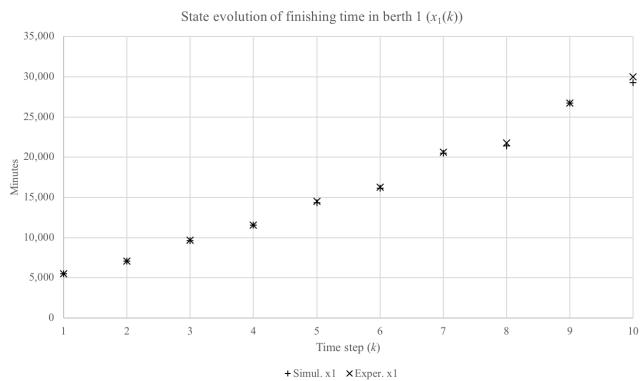


Fig. 6. The plot of trajectory of state variable  $x_1(k)$  that describes the finishing time of the 1<sup>st</sup> berth position. The trajectory that is based on the predictive model (6) – (14) is shown in + while that obtained from the actual field experiment is shown in x. The abscissa is the event-time  $k$  and the ordinate is the value of the state variable at each event-time.

framework can capture the discrete-event dynamics of multiple berthing positions and multiple quay cranes, as well as, asynchronous berthing time for different berthing positions. The discrete-event model has been used in the development of a real-time model predictive allocation strategy for solving the I-BCAP. We have also presented several pre-conditioning steps to tractably solve the optimization problem. Simulation results have shown the efficacy of our proposed method where Monte Carlo simulations have been performed using large-scale realistically-generated datasets.

We have also conducted a field experiment in an actual operational container terminal where our proposed method was applied. The results show that our proposed method still outperforms both the standard and the state-of-the-art methods (based on the validated model).

For future works, our modeling framework can be generalized to the complete seaport operations and use it for real-time optimization of the integrated seaport operations. The movement of QCs, which is not incorporated yet in the current model, is also an interesting part of the future works. The movement will make the models more realistic. Another area of interests is the real implementation of our proposed method that incorporates the dissatisfaction level of shipping liners in the optimization. It has been observed during the field experiment period that some of the arriving ships had shown their dissatisfaction due to a possible uncommon schedule, i.e.

a ship may be berthed after another ship which arrive later after the former vessel. Therefore, taking into account the shipping liners' dissatisfaction in the cost function can be an interesting and relevant future work.

#### ACKNOWLEDGMENT

The authors gratefully acknowledge Dana Amin, Andi Isnovandiono, and Al Imran La Ode from Indonesia Port Corporation for their contribution in providing the data and for their information about practical aspects in container terminal operation.

#### REFERENCES

- [1] A. Alessandri, C. Cervellera, M. Cuneo, M. Gaggero, and G. Soncin, "Modeling and feedback control for resource allocation and performance analysis in container terminals," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 4, pp. 601–614, 2008.
- [2] A. Alessandri, C. Cervellera, and M. Gaggero, "Predictive control of container flows in maritime intermodal terminals," *IEEE Trans. Contr. Syst. Techn.*, vol. 21, no. 4, pp. 1423–1431, 2013.
- [3] C. Bierwirth, and F. Meisel, "A survey of berth allocation and quay crane scheduling problems in container terminals," *Eur. J. Oper. Res.*, vol. 202, pp. 615–627, 2016.
- [4] J. Blazewicz, T. C. E. Chang, M. Machowiak, and C. Oguz, "Berth and quay crane allocation: a moldable task scheduling model," *Journ. Oper. Res. Soc.*, vol. 62, pp. 1189–1197, 2011.
- [5] F. B. van Boetzelaer, *Model predictive scheduling for container terminals*, M.S. thesis, Delft University of Technology, Delft, the Netherlands, 2013.
- [6] C. Caballini, C. Pasquale, S. Sacone, and S. Siri, "An event-triggered receding-horizon scheme for planning rail operations in maritime terminals," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 365–375, 2014.
- [7] R. T. Cahyono, E. J. Flonk, and B. Jayawardhana, "Dynamic berth and quay crane allocation for multiple berth positions and quay cranes," *Proc. 14th IEEE Eur. Contr. Conf.*, Linz, 2015.
- [8] R. T. Cahyono, E. J. Flonk, and B. Jayawardhana, "Supplemental Material: Discrete-event systems modeling and model predictive allocation algorithm for integrated berth and quay crane allocation," [online] Available at: [https://www.rug.nl/research/portal/files/74419315/Appendix\\_BCAP\\_paper.pdf](https://www.rug.nl/research/portal/files/74419315/Appendix_BCAP_paper.pdf)
- [9] H. J. Carlo, I. F. A. Vis, and K. J. Roodbergen, "Seaside operations in container terminals: literature review, trends, and research directions," *Flex. Serv. Manuf. J.*, vol. 27, no. 2, pp. 224–262, 2015.
- [10] C.G. Cassandras, S. Lafortune, *Introduction to Discrete Event Systems*, 2nd edition, Springer, New York, 2008.
- [11] D. Chang, Z. Jiang, W. Yan, and J. He, "Integrating berth allocation and quay crane assignments," *Transp. Res. Part E*, vol. 46, pp. 975–990, 2010.
- [12] R. David and H. Alla, "Petri nets for modeling of dynamic systems," *Automatica*, vol. 30, no. 2, pp. 175–202, 1994.
- [13] B. De Schutter, and T. v.d. Boom, "MPC for discrete-event systems with soft and hard synchronization constraints," *Int. J. Con.*, vol. 76, no. 1, pp. 82–94, 2007.
- [14] G. Giallombardo, L. Moccia, M. Salani, and I. Vacca, "Modeling and solving the tactical berth allocation problem," *Transp. Res. Part B*, vol. 44, pp. 232–245, 2010.
- [15] M. M. Goliass, M. Boile, and S. Theofanis, "A lamda-optimal based heuristic for the berth scheduling problem," *Transp. Res. Part C*, vol. 18, pp. 794–806, 2010.
- [16] Y. Guan, W. Xiao, R.K. Cheung, and C. Li, "A multiprocessor task scheduling model for berth allocation: heuristic and worst-case analysis," *Oper. Res. Lett.*, vol. 30, pp. 343–350, 2002.
- [17] P. Hansen, C. Oğuz, and N. Mladenović, "Variable neighborhood search for minimum cost berth allocation," *Eur. J. Oper. Res.*, vol. 191, pp. 636–649, 2008.
- [18] H-P. Hsu, "A HPSO for solving dynamic and discrete berth allocation problem and dynamic quay crane assignment problem simultaneously," *Swar. Evol. Com.*, vol. 27, pp. 156–168, 2016.
- [19] A. Imai, E. Nishimura, and S. Papadimitriou, "The dynamic berth allocation problem for a container port," *Transp. Res. Part B*, vol. 35, pp. 401–417, 2001.
- [20] A. Imai, H. C. Chen, E. Nishimura, and S. Papadimitriou, "The simultaneous berth and quay crane allocation problem," *Transp. Res. Part E*, vol. 44, pp. 401–417, 2008.
- [21] A.I. La Ode, Department of Operations, Indonesia Port Corporation, private communication, 2016.
- [22] F. Meisel and C. Bierwirth, "Heuristics for the integration of crane productivity in the berth allocation," *Transp. Res. Part E*, vol. 45, pp. 196–209, 2009.
- [23] E. Nishimura, A. Imai, and S. Papadimitriou, "Berth allocation planning in the public berth system by genetic algorithms," *Eur. J. Oper. Res.*, vol. 131, pp. 282–292, 2001.
- [24] B. Raa, W. Dullaert, and R. van Schaeren, "An enriched model for the integrated berth allocation and quay crane assignment problem," *Expert Systems with Applications*, vol. 38, pp. 14136–14147, 2011.
- [25] R. Stahlbock and S. Voß, "Operations research at container terminals: a literature update," *OR Spectrum*, vol. 30, pp. 1–52, 2007.
- [26] D. Steenken, S. Voß, and R. Stahlbock, "Container terminal operation and operations research - a classification and literature review," *OR Spectrum*, vol. 33, pp. 3–49, 2004.
- [27] R. Tavakkoli-Moghaddam, A. Makui, S. Salahi, M. Bazzazi, and F. Taheri, "An efficient algorithm for solving a new mathematical model for a quay crane scheduling problem in container ports," *Comp. Ind. Eng.*, vol. 56, pp. 241–248, 2009.
- [28] C. Zhang, L. Zheng, Z. Zhang, L. Shi, and A. J. Armstrong, "The allocation of berths and quay cranes by using a sub-gradient optimization technique," *Comp. Ind. Eng.*, vol. 58, pp. 40–50, 2010.

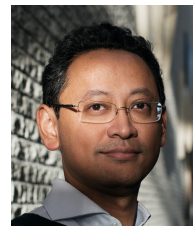


for complex systems, for examples are seaport operations, systems logistics, and industrial systems.



**Rully Tri Cahyono** received the B.Sc. degree in industrial engineering, in 2009, and the M.Sc. degree in industrial engineering and management, in 2011, from Institut Teknologi Bandung, Bandung, Indonesia. Currently, he is a Ph.D student in the research group of Discrete Technology and Production Automation, Faculty of Science and Engineering, University of Groningen, Groningen, The Netherlands. He is also a junior lecturer in Department of Industrial Engineering, Institut Teknologi Bandung. His research interests are on the mathematical modeling

**Engel Jacob Flonk** received the B.Sc. degree in industrial engineering, in 2012, and the M.Sc. degree in industrial engineering and management, in 2014, from University of Groningen, The Netherlands. Currently, he is a logistics engineer in Brink Tower Systems, The Netherlands. His responsibility is with the planning of the logistics activities in the company. Previously, he was a transport planner in Holtrop - Van der Vlist.



**Bayu Jayawardhana** (SM13) received the B.Sc. degree in electrical and electronics engineering from the Institut Teknologi Bandung, Bandung, Indonesia, in 2000, the M.Eng. degree in electrical and electronics engineering from the Nanyang Technological University, Singapore, in 2003, and the Ph.D. degree in electrical and electronics engineering from Imperial College London, London, U.K., in 2006. Currently, he is a full professor in the Faculty of Science and Engineering, University of Groningen, Groningen, The Netherlands. He was with Bath University, Bath, U.K., and with Manchester Interdisciplinary Biocentre, University of Manchester, Manchester, U.K. His research interests are on the analysis of nonlinear systems, systems with hysteresis, mechatronics and smart transportation systems. He is a Subject Editor of the International Journal of Robust and Nonlinear Control, an associate editor of the European Journal of Control and of the IEEE Transactions on Control Systems Technology, and a member of the Conference Editorial Board of the IEEE Control Systems Society.